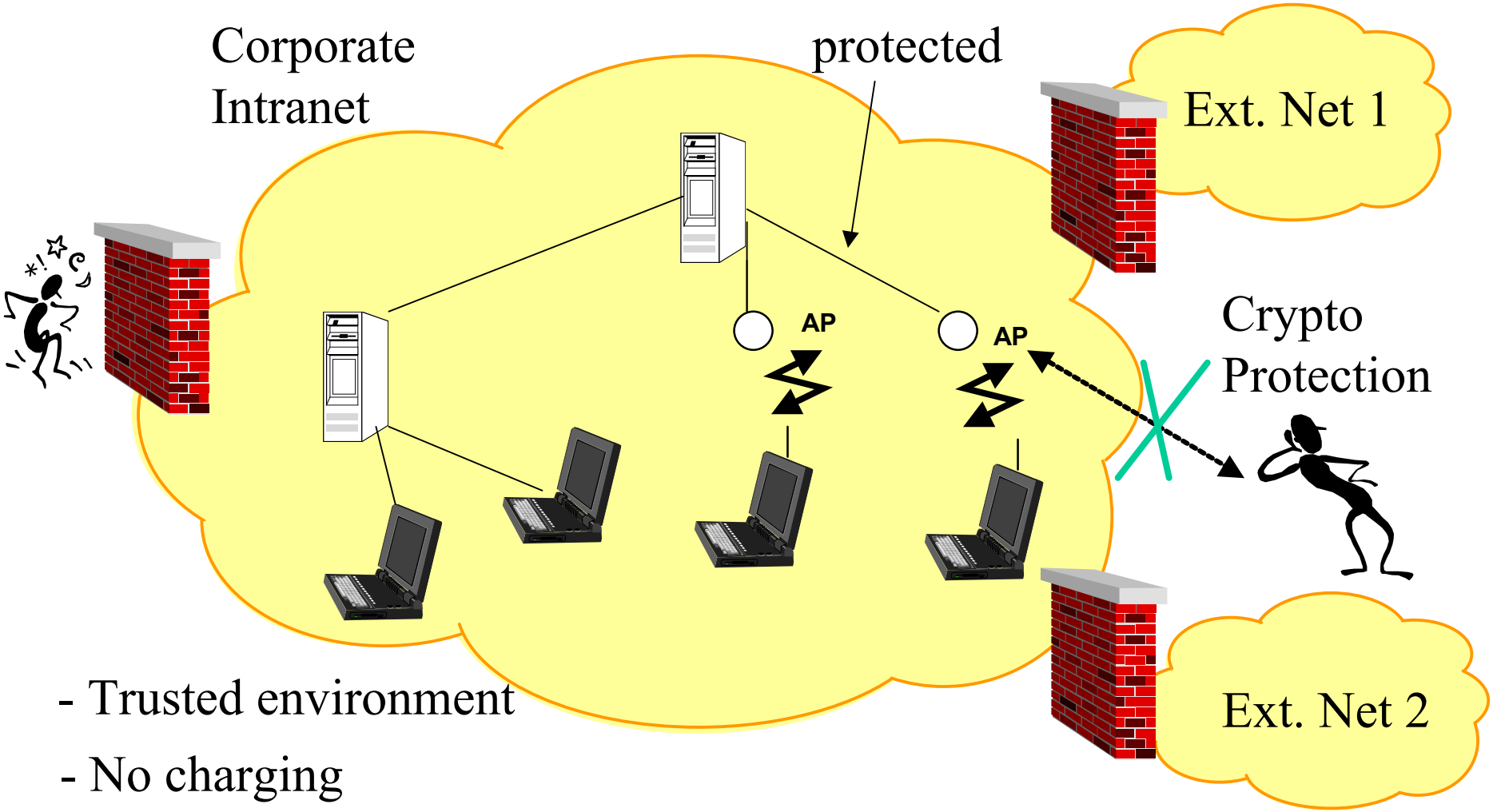
Security issues in public access LAN architectures

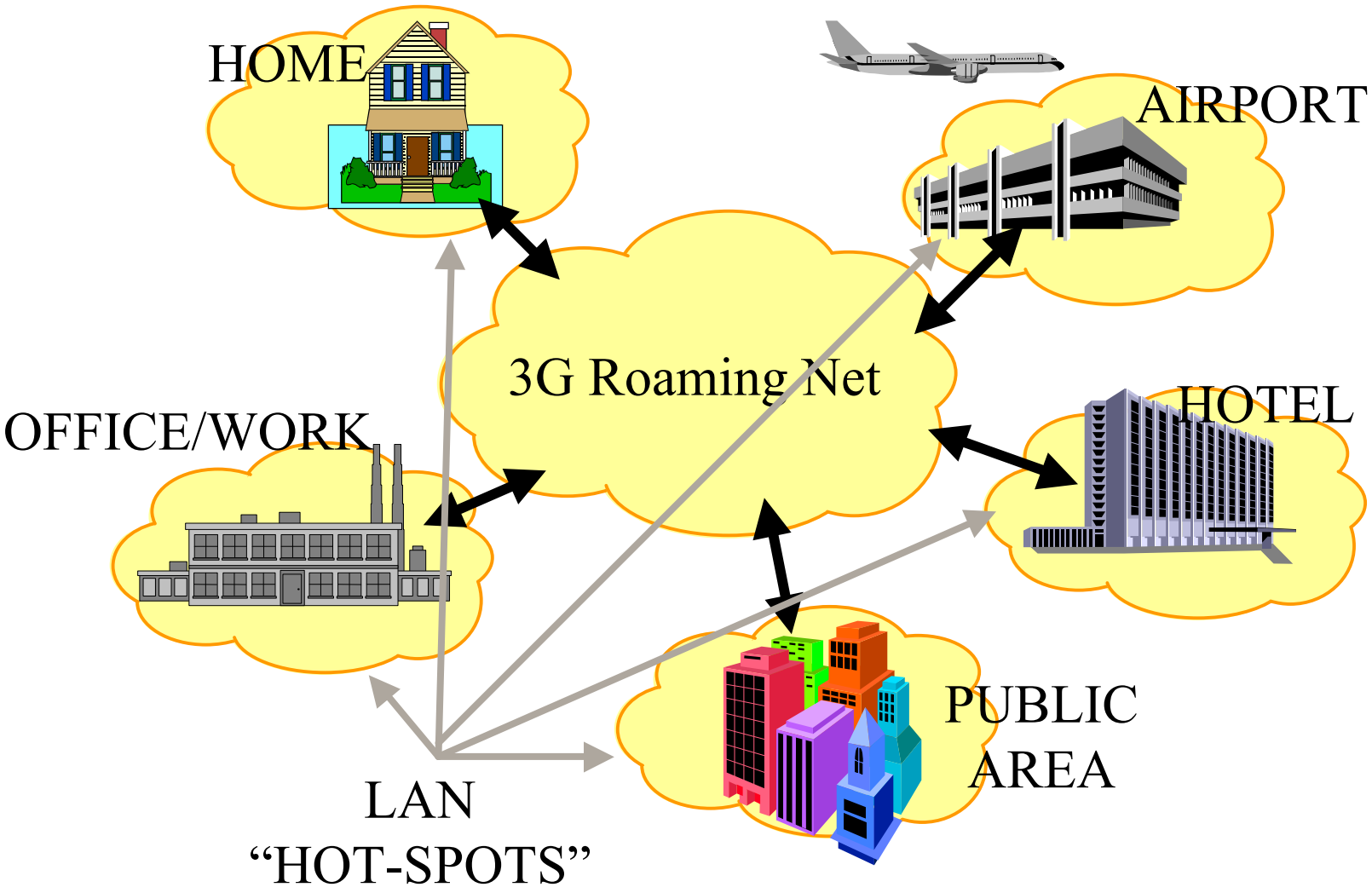
Mats Näslund, Stefan Rommer
Ericsson

Example: Traditional LAN Environment

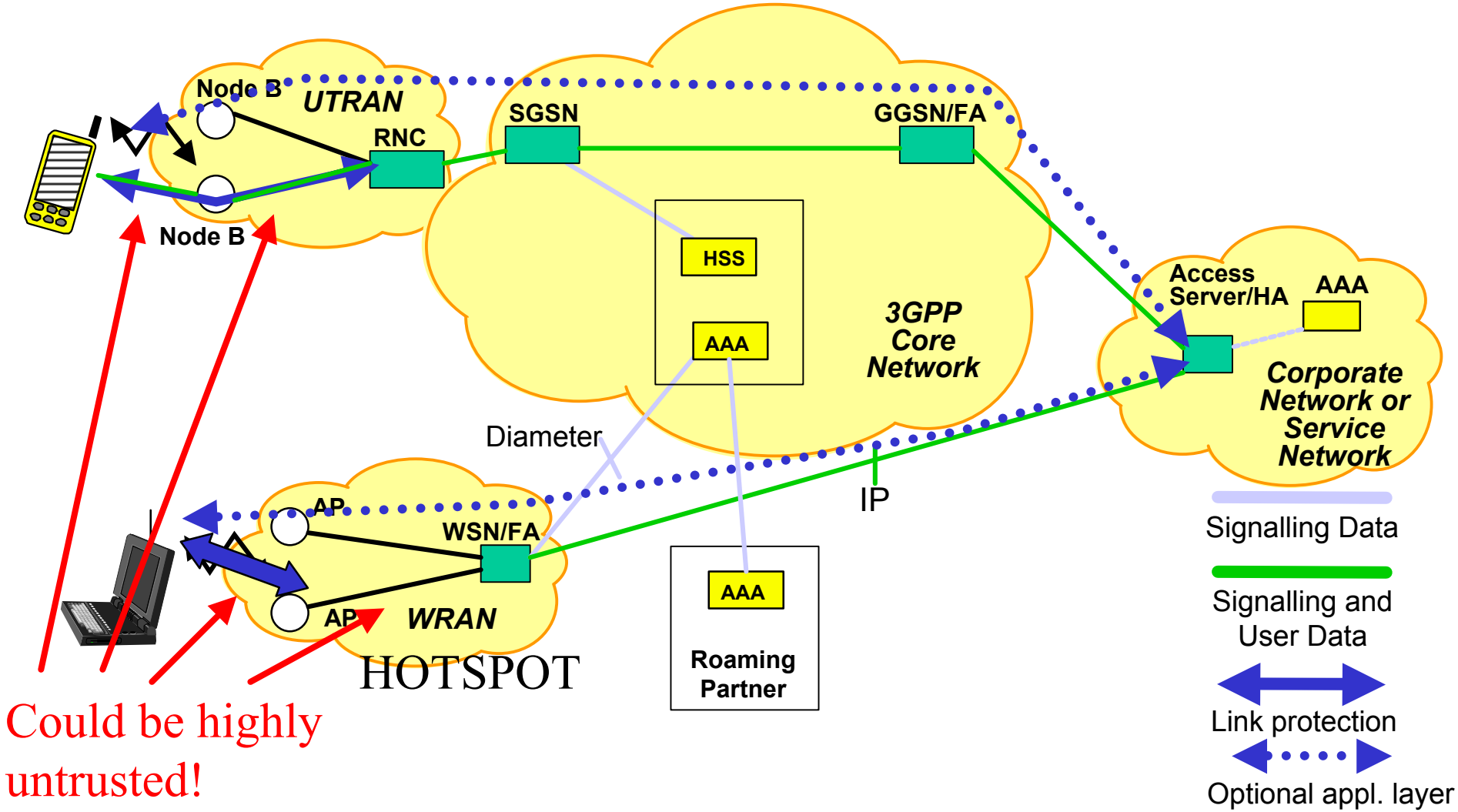


- Trusted environment
- No charging

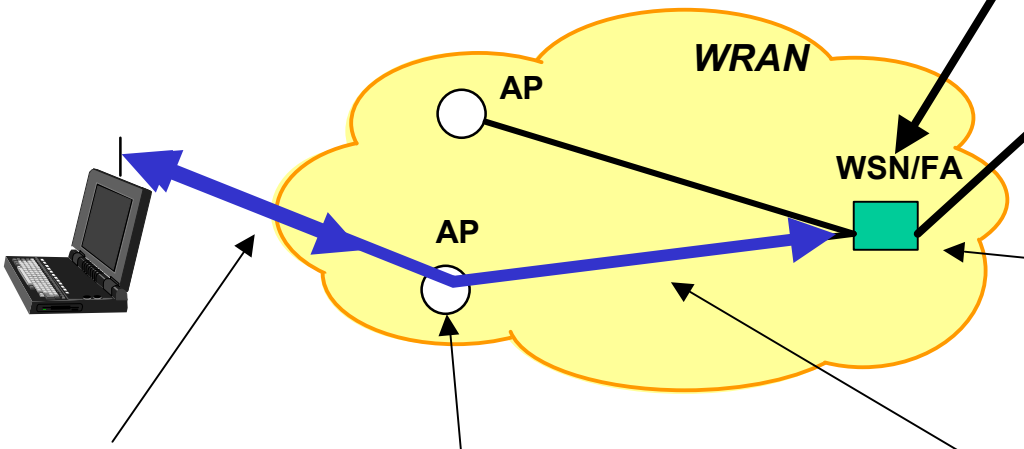
Public Access LAN Environments



Access Scenario



Security End-points



Billing data collected here

Protected! (Physically secured)

Security exists in *some* 802 standards

AP, bridge,..etc in public environment could be tampered with:
End-to-end desired

No protection at all!
Robust charging requires *at least* integrity protection.

(W)LAN Public Access

Business aspects

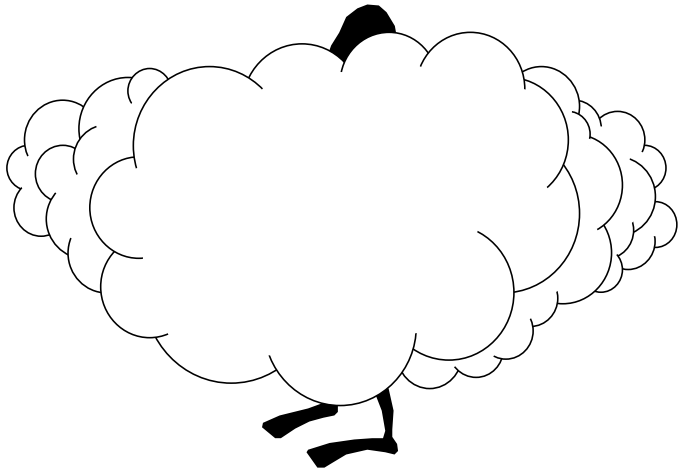
- Mobile Operators see WLAN Public access as important scenario
- Highly desirable to match operators' existing security/threat model

Technical aspects

- Other forms of 802-based broadband access emerging in operators product portfolio, uniform treatment of security facilitates e.g. seamless roaming between accesses (e.g. 802.11 $\Leftrightarrow$ 802.16)

Note Well:

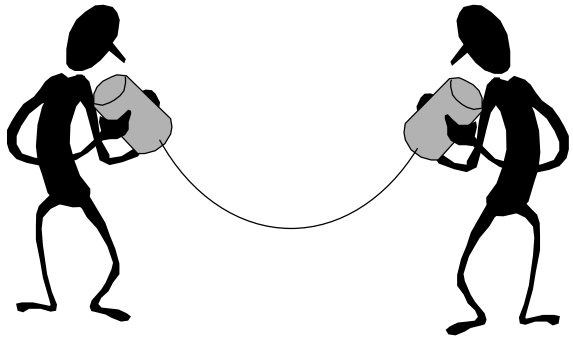
Charging is *not* the only issue, users' privacy threatened without proper protection in public environment!



Top on “Wishlist”

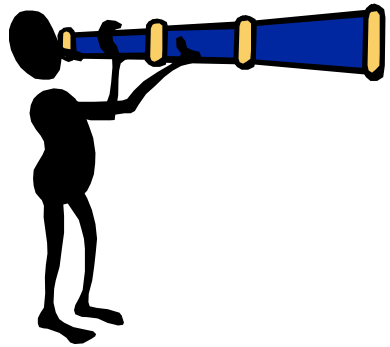
A new 802 link security should enable *end-to-end traffic protection* (at least integrity) between mobile station and trusted ”access server node”

802 link security should be *uniform* and *access independent* (transparent to bridges etc)

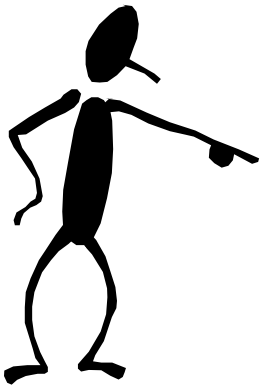


Other Items for Study

Security for Access Network Discovery



Security for Access Network Selection



To Consider

Security problems here are due to use of a std in setting (e.g. public access) for which it wasn't originally designed (e.g. corporate intranet)

Be open to new use-cases/threats!

E.g. in the future may be *very weak* trust between operator and access network provider, repudiation of user auth. might also be required for robust charging

Other Opinions

Make it plug-and-play: defined down to crypto transform level, not an abstract framework

Learn from previous misstakes (e.g. *no* unkeyed MICs like in WEP)

While we're at it, fix sub-optimal features from other 802 stds (e.g. the consequent but wrong "authenticate-then-integrity" processing order)